1. Description

AWK is a programming language designed for text processing and typically used as a data extraction and reporting tool. It is a standard feature of most Unix-like operating systems. (From Wikipedia)

2. Useful Links/sources

- Wikipedia article
- sed vs AWK (stackoverflow)
- Official GAWK tutorial
- Get started with GAWK: AWK language fundamentals (PDF)

3. Some General Basics

- Created by Alfred **A**ho, Peter **W**einberger, and Brian **K**ernighan
- A file is treated as a sequence of records, and by default each line is a record. AWK reads the input a line at a time.
- Each record can be broken down further into individual chunks called fields (Both images from 'Get started with GAWK: AWK language fundamentals (PDF)')
- Most widely used implementation: GAWK (GNU AWK), installed on most GNU Linux Systems.
- main rival of AWK: Perl.

4. Example Files

Download example file 1

Download file homo.gtf

```
wget http://bioinformatics.med.uoc.gr/bioinfo_grad/homo.gtf
```

This is an abridged version of:

ftp://ftp.ensembl.org/pub/release-86/gtf/homo_sapiens/Homo_sapiens.GRCh38.86.gtf.gz

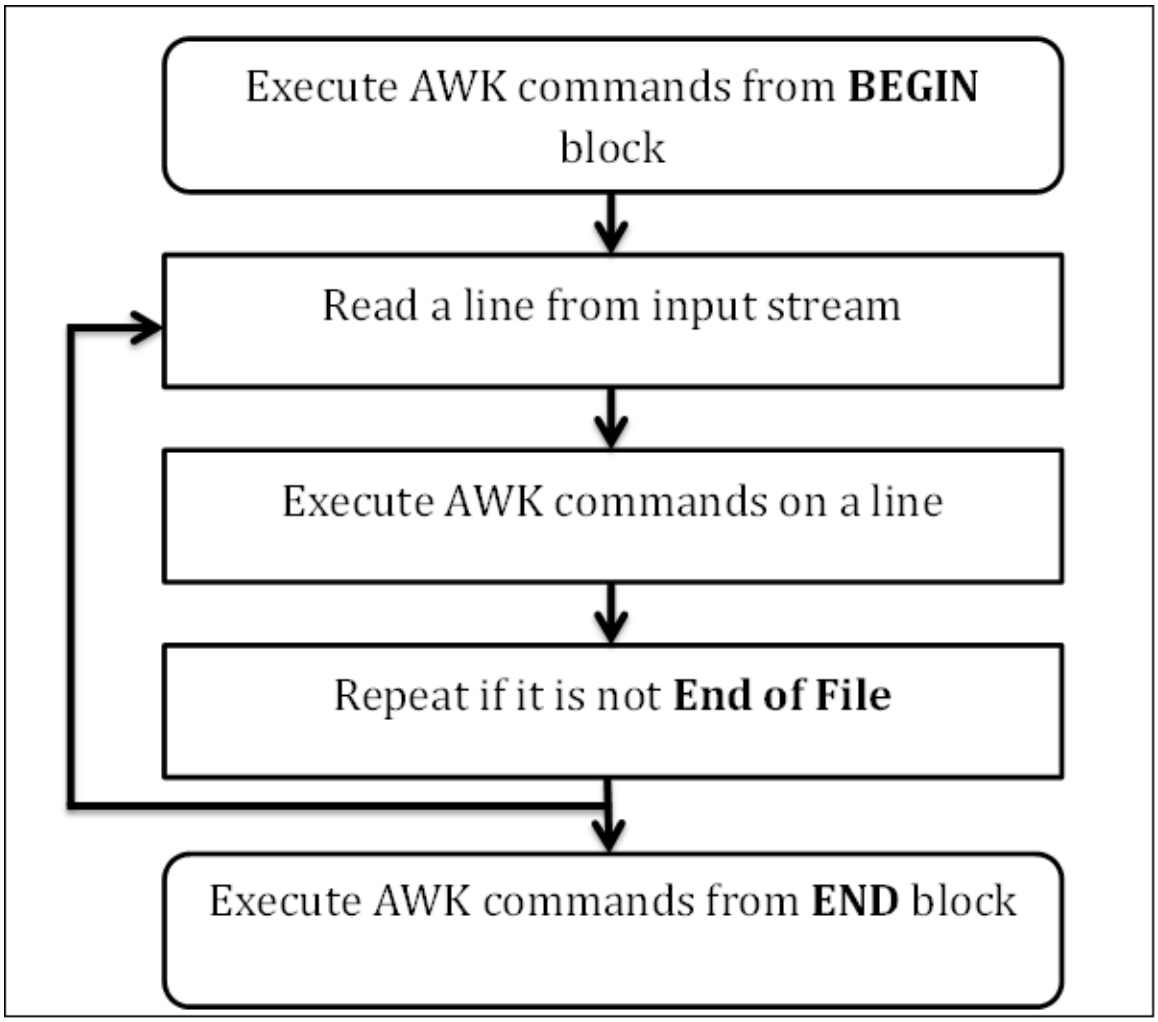
GTF Files

The GFF (General Feature Format) format consists of one line per feature, each containing 9 columns of data, plus optional track definition lines.

Format Description: GFF/GTF File Format

5. Let's start

AWK Workflow



https://www.tutorialspoint.com/awk

Basic structure

Basic format:

pattern { action }

```
gawk '/pattern/'
gawk '/pattern/ {action}'
```

Full format:

BEGIN {do stuff once at the beginning} pattern {action} END {do stuff once at the end}

The Less command

```
less <FILE>
less -S <FILE>
```
The -S argument disables word wrapping

Basic examples (print, conditions, piping)

- print all lines

```
gawk '{print}' homo.gtf
```

- print specific column(s)

```
gawk '{print $1}' homo.gtf
gawk '{print $1,$2}' homo.gtf
gawk '{print "column 1:"$1," column 3:"$3}' homo.gtf
```

- print when condition is met, also wrap

```
gawk '$1=="X"' homo.gtf
gawk '$1=="X"' homo.gtf | less -S
gawk '$1 ~ /^X/' homo.gtf | less -S
```

- count instances where first column condition is met

```
gawk '$1=="1"' homo.gtf |wc -l
more homo.gtf | gawk '$1=="1"' | wc -l
```

- print specific column(s) based on a condition

```
gawk '$1=="1" {print $3}' homo.gtf | less -S
```

- print to a file

```
gawk '$1=="1" {print $3}' homo.gtf | less -S > results.txt
```

Running AWK program saved in a file

1. write this program to a file

```
{ print }
```

make it executable and run the program this way:

```
gawk -f program_name homo.gtf
```

2. write this program to a file

```
#!/usr/bin/gawk -f
{ print }
```

run the program this way:

```
./program_name homo.gtf
```

The BEGIN and END patterns

BEGIN pattern specifies actions to be performed before any records are processed:

```
BEGIN {action}
```

END pattern specifies actions to be performed after all records are processed

```
END {action}
```

Example program:

```
#!/usr/bin/gawk -f
BEGIN {
 print "Start\n-----------------"
}
$1=="1" {print $3}
END {
 print "------------";
 print "End"
}
```

Built-in Variables

1. FS: Field Separator

Type all users from /etc/paswwd:

```
cat /etc/passwd | gawk 'BEGIN{FS=":"}{print $1}'
```

```
cat /etc/passwd | gawk -F: '{print $1}'
```

2. RS: Record Separator

3. OFS: Output Field Separator

```
cat /etc/passwd | gawk 'BEGIN{FS=":";} {print $1,$7}'
```

```
cat /etc/passwd | gawk 'BEGIN{FS=":"; OFS=" - "}{print $1,$7}'
```

Variables

```
gawk '{++cnt} END {print cnt}' homo.gtf
gawk '{print $1;++counter} END {print counter}' homo.gtf
```